2023-12-24

王英泰 1120210964

北京理工大学计算机学院

计算机病毒防治

实验五 Windows恶意程序实验报告

目录

[实验准备 2](#_Toc156226178)

[一、 实验环境和所用到的软件 2](#_Toc156226179)

[二、 基本知识 2](#_Toc156226180)

[Lab 7-1 5](#_Toc156226181)

[一、 当计算机重启后，这个程序如何确保它继续运行 5](#_Toc156226182)

[二、 为什么这个程序会使用一个互斥量 8](#_Toc156226183)

[三、 检测这个程序的基于主机的特征是什么 8](#_Toc156226184)

[四、 检测这个恶意代码的基于网络特征是什么 9](#_Toc156226185)

[五、 这个程序的目的是什么 9](#_Toc156226186)

[六、 这个程序什么时候完成执行 11](#_Toc156226187)

[Lab 7-2 12](#_Toc156226188)

[一、 这个程序如何完成持久化驻留 12](#_Toc156226189)

[二、 这个程序的目的是什么 12](#_Toc156226190)

[三、 这个程序什么时候完成执行 15](#_Toc156226191)

[Lab 7-3 15](#_Toc156226192)

[一、 这个程序如何完成持久化驻留使得重启后能够继续运行 15](#_Toc156226193)

[二、 这个恶意代码的两个明显的基于主机的特征是什么 19](#_Toc156226194)

[三、 这个程序的目的是什么 20](#_Toc156226195)

[四、 一旦这个恶意代码被安装，该如何移除它 22](#_Toc156226196)

[实验总结 23](#_Toc156226197)

# 实验准备

1. 实验环境和所用到的软件

* Windows 10 x86
* Procmon、IDA Pro等

1. 基本知识
2. 文件系统函数

* CreateFile
* ReadFile、WriteFile
* CreateFileMapping：创建一个内存映射文件内核对象
* MapViewOfFile：将一个文件映射对象映射到当前应用程序的地址空间

1. 特殊文件
   * 共享文件：访问存储在网络上的共享文件夹或文件
2. 文件名以\\serverName\share或\\?\serverName\share开始
3. \\?\前缀禁止所有字符串解析，允许访问更长的文件名
   * 名字空间
4. 名字空间可以被认为是固定数量的文件夹，每个文件存储不同类型的对象
5. 最低级的名字空间是NT名字空间，有前缀\，该名字空间可以访问所有设备，所有其他名字空间都位于该名字空间下
   * 备用数据流ADS特性允许附加数据被添加到一个已存的NTFS文件中，相当于添加一个文件到另外一个文件中：File.txt:otherfile.txt
6. 注册表：用于存储系统或程序的配置信息
   * Root keys

文本

描述已自动生成

1. HKEY\_LOCAL\_MACHINE：存储本地机器的全局设置
2. HKEY\_CURRENT\_USER：对于当前用户的特别设置
3. HKEY\_CLASSES\_ROOT：定义类型的信息
4. HKEY\_CURRENT\_CONFIG：存储当前硬件配置，尤其是当前和标准配置的不同之处
5. HKEY\_USERS：对默认用户、新用户和当前用户定义设置

* Subkey
* Key：根键和子键都是键
* Value entry：值项，由name和data组成
* 计算机\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run、计算机\HKEY\_LOCAL\_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run：存储开机自启软件
  1. 可以使用Autoruns软件来查看
* 常见的注册表函数

1. RegOpenKeyEx：打开注册表项来进行编辑和查询，有些函数不需要打开就可以进行编辑或查询
2. RegSetValueEx：添加一个新的注册表项并设置值
3. RegGetValue：获取注册表项的值

* 双击.reg文件可以直接修改注册表，可被恶意软件利用
* 恶意软件通常修改注册表实现开机自启

1. 网络
   * socket
     1. socket函数主要在ws2\_32.dll库中
     2. 服务端的函数调用：socket、bing、listen、accept、send、recv
     3. 客户端的函数调用：socket、connent、send、recv
   * WinINet：高水平的API，应用级的接口
     1. 位于wininet.dll库中
     2. InternetOpen：初始化网络连接
     3. InternetOpenUrl：可以用来链接URL
     4. InternetReadFile：读取从网络上下载的文件数据
2. 恶意软件的执行流改变
   * 跳转
   * 调用
   * 使用DLL：执行外部代码
     1. 一些恶意软件附着到其他进程，但每个进程只能包含一个.exe文件。恶意软件有时会使用dll将自己加载到另一个进程中
   * 创建一个新进程
     1. 使用CreateProcess来创建一个简单的远程shell
     2. 恶意软件通常会将一个程序存储在资源部分来创建一个新进程。当程序运行时，它将从PE头中提取额外的可执行文件，将其写入磁盘，然后调用CreateProcess来运行程序。这也可以通过DLL和其他可执行代码来完成。此时需要使用Resource Hacker来提取恶意程序

# Lab 7-1

1. 当计算机重启后，这个程序如何确保它继续运行

使用IDA Pro打开Lab07-01.exe，并切换到Imports窗口查看这个程序的导入。

图形用户界面, 应用程序, 表格

描述已自动生成

从ADVAPI32库中导入的OpenSCMananger、CreateService、StartServiceCtrlDispatcherA函数表明该程序可能与服务有关。从OpenSCManagerA和CreateServiceA函数可以推测出该恶意代码可能会利用服务控制管理器创建一个新服务；StartServiceCtrlDispatcherA函数将服务进程的main线程连接到服务控制管理器，这会使线程成为调用进程的服务控制调度程序线程，这说明该恶意代码确实是个服务。

接下来我们分析主函数，主函数调用StartServiceCtrlDispatcherA函数。通过查看MSDN，我们可以得知函数的参数是一个指向SERVICE\_TABLE\_ENTRY结构的数组的指针。SERVICE\_TABLE\_ENTRY结构体中有两个成员：服务名和服务处理程序。

文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

从参数可以看出恶意代码安装成的服务名应为“MalService”，指定的服务控制函数为sub\_401040，该子函数会在执行StartServiceCtrlDispatcherA后被调用，双击跳转。

图片包含 日程表

描述已自动生成

这段代码首先调用OpenSCManager打开一个服务控制管理器的句柄，然后调用GetCurrentProcess获取当前进程的伪句柄，紧接着调用GetModuleFileName函数，并传入刚获取的恶意代码进程伪句柄，从而获取恶意代码的全路径名，这个全路径名被传入CreateServiceA函数，从而将该恶意代码安装成一个名为“Malservice”的服务。从CreateServiceA的dwStartType并对照MSDN，可以知道该程序在创建服务时设置为了开机自启。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. 为什么这个程序会使用一个互斥量

找到与互斥量相关的代码如下：

图形用户界面, 应用程序

描述已自动生成

从上图可以看出，OpenMutexA会尝试打开名为“HGL345”的互斥量，如果打开成功则程序退出，否则创建一个名为“HGL345”的互斥量 。

综上，恶意代码使用互斥量是为了确保该计算机上只有一个恶意代码实例在运行。

1. 检测这个程序的基于主机的特征是什么

由前两问分析可知有两个基于主机特征：名为“Malservice”的服务、名为“HGL345”的互斥量。

1. 检测这个恶意代码的基于网络特征是什么

从IDA Pro的导入窗口可以看到该程序导入了InternetOpenUrlA和InternetOpenA函数。



双击跳转到函数在代码中位置，然后按下“X”键打开交叉引用，找到调用函数的位置。

图形用户界面, 应用程序

描述已自动生成

可以看到InternetOpenA函数的szAgent参数，即使用的User-Agent为Internet Explorer 8.0，而InternetOpenUrlA函数要访问的地址是http://www.malwarenanlysisbook.com。这两个就是该恶意代码基于网络的特征。

我们注意到这个函数中有一个无条件跳转指令，即程序会一直调用InternetOpenUrlA来下载主页。

1. 这个程序的目的是什么

继续分析在sub\_401040中的程序，我们发现接下来调用了SystemTimeToFileTime函数，通过查阅MSDN得知该函数将系统时间转换为文件时间格式，而系统时间基于UTC。即该函数将系统时间2100年1月1日0点0分0秒转换为文件时间。

文本

描述已自动生成

将上述时间点转换为文件时间类型后，先调用了CreateWaitableTimerA函数创建定时器对象，然后调用SetWaitableTimer函数设置定时器，其中参数lpDueTime为上面转换的文件时间结构体，最后调用WaitForSingleObject函数等待计时器对象变为有信号状态或等待时间达到FFFFFFFFh毫秒（这当然是达不到的时间），也就是说会等到2100年1月1日0点然后函数返回继续往下执行。

如果在等待期间出错，则WaitForSingleObject函数返回非0值，如果定时器时间到则返回0。

如果计时器时间到，则开始执行如下代码：开始一个循环，创建20个线程，每个线程执行StartAddress函数。

图形用户界面, 应用程序

描述已自动生成

综上所述，这个恶意代码自己安装成一个自启动服务，保证只要计算机开启，恶意代码就在运行，然后等到2100年1月1日0点，开启20个线程无限次持续访问http://www.malwarenanlysisbook.com，该恶意代码可能是用来对目标网址进行DDoS攻击。

1. 这个程序什么时候完成执行

从上述分析可以得出，这个程序永远不会停止，它等待一个计时器直到2100年，如果时间到则进入无限循环。

# Lab 7-2

1. 这个程序如何完成持久化驻留

使用IDA Pro打开Lab07-02.exe，结果如下：

图形用户界面, 应用程序

描述已自动生成

通过左侧的Functions窗口可以看到很少的函数，没有与持久化驻留相关的代码，因此这个恶意代码没有实现持久化驻留。

1. 这个程序的目的是什么

对\_main函数中的代码进行分析。该程序首先调用了Oleinitialize函数，该函数初始化当前单元上的 COM 库，将并发模型标识为单线程单元 (STA) ，并启用以下备注部分中介绍的其他功能。 应用程序必须先初始化 COM 库，然后才能调用 CoGetMalloc 和内存分配函数以外的 COM 库函数。

图片包含 Teams

描述已自动生成

然后调用了CoCreateInstance函数来创建并默认初始化与指定 CLSID 关联的类的单个对象。返回值被IDA重命名为ppv。

文本

描述已自动生成

接下来查看类型标识符CLSID和接口表示符IID，可得到类型标识符CLSID值为0002DF01-0000-0000-C000-000000000046，这代表Inter Explorer；接口标识符IID值为D30C1661-CDAF-11D0-8A3E-00C04FC9E26E，这代表IWebBrowser2接口。

图形用户界面, 文本

中度可信度描述已自动生成

之后调用VariantInit函数初始化变量，再调用SysAllocString函数为字符串“http://www.malwareanalysisbook.com/ad.html”分配内存。

Eax中存储ppv，ppv指向了COM对象的位置，mov edx,[eax]指令使edx指向COM对象基地址，call dword ptr [edx+2Ch]调用了IWebBrowser2接口偏移0x2Ch即44处的函数，每个函数地址占4字节，也就是调用了序号11即第12个函数。这个函数是Navigate函数，它允许一个程序启动Internet Explorer并访问一个Web地址，而压入的参数esi中保存着调用SysAllocString分配了内存空间的字符串<http://www.malwareanalysisbook.com/ad.html>。

图形用户界面, 文本, 应用程序

描述已自动生成

访问完该网址后，程序正常退出。也就是说该恶意代码的目的就是访问一个网址，而从网址来看是一个广告页面。



1. 这个程序什么时候完成执行

从上述分析中可以看出，该程序在显示完广告页面后就会退出。

# Lab 7-3

1. 这个程序如何完成持久化驻留使得重启后能够继续运行

使用IDA Pro打开Lab07-03.exe。main函数首先比较argc是否为2，如果不为2则程序退出。

图形用户界面, 应用程序

描述已自动生成

然后程序比较命令行参数是否为“WARNING\_THIS\_WILL\_DESTROY\_YOUR\_MACHINE”，如果不是则程序立即退出。

图示

中度可信度描述已自动生成

之后，main函数调用了CreateFile、CreateFileMapping和MapViewOfFile，它打开了文件C:\\Windows\\System32\\Kernel32.dll和Lab07-03.dll，并将C:\\Windows\\System32\\Kernel32.dll和Lab07-03.dll映射到内存。

表格

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

之后，main函数调用了sub\_401040和sub\_401070。查看这两个函数，我们发现这两个函数只是进行了简单的计算，并且都调用sub\_401000函数。我们可以猜测这两个函数可能是对刚才映射到内存中文件进行修改。

之后，main函数调用了CloseHandle来关闭文件句柄，接着调用CopyFileA函数将Lab07-03.dll复制到C:\\windows\\system32\\kerne132.dll。接着main函数调用了sub\_4011E0函数。

图形用户界面, 文本, 应用程序

描述已自动生成

接下来我们分析函数sub\_4011E0，该函数的功能是搜索C盘所有后缀名为exe的文件。如果后缀名为exe则调用sub——4010a0函数。该函数首先调用CreateFile、CreateFileMapping和MapViewOfFile来映射整个文件。

文本

描述已自动生成

图片包含 文本

描述已自动生成

之后，该函数搜索exe文件中所有kernel32.dll并将其更换为kerne132.dll。

进入内存。

综上，该程序将Lab07-03.dll复制到C:\Windows\System32\kerne123.dll，然后搜索C盘上所有后缀名为exe的文件，并将其中kernel32.dll并将其更换为kerne132.dll，从而实现持久化。

1. 这个恶意代码的两个明显的基于主机的特征是什么

在上面的分析中，该恶意软件会创建文件C:\Windows\System32\kerne123.dll ，同时也会创建信号量“SADFHUHF”来保证任何时候只有一个恶意软件在运行。

1. 这个程序的目的是什么

使用IDA Pro来打开Lab07-03.dll。先来看一下函数调用图。可以看到首先调用了\_\_alloa\_probe来在栈上分配空间。

图片包含 文本

描述已自动生成

然后调用了OpenMutexA和CreateMutexA函数来创建信号量”SADFHUHF”,该信号量的作用是保证只有一个恶意软件实例在系统中运行。

文本

描述已自动生成

之后使用socket在本机和127.26.152.13之间建立连接，端口号为80。被感染机器向远程机器发送“hello”字符串。

图示

低可信度描述已自动生成

在调用recv函数收到远程机器的数据后，该程序使用strncmp函数来比较收到的字符串是否以“sleep”开头，如果是则调用Sleep函数休眠60s。如果收到的字符串以“exec”开头，则会调用CreateProcessA函数来创建一个新进程。接着我们分析CreateProcessA函数的输入参数lpCommandLine是从远程主机上接受到的。

图片包含 图形用户界面

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

综上，该程序会创建一个后门，它连接到远程主机并且接受两个命令“sleep”和“exec”。

1. 一旦这个恶意代码被安装，该如何移除它

这个程序很难删除，因为它会感染系统上的每个.exe文件。在这种情况下，最好从备份中恢复。如果从备份中恢复特别困难，可以保留恶意kerne132.dll文件并对其进行修改以删除恶意内容。也可以复制kernel32.dll并将其命名为kerne132.dll，或者编写一个程序来撤消对PE文件的所有更改。

# 实验总结

本章介绍了对恶意软件分析非常重要的Windows概念，如进程、线程和网络、注册表。除了分析恶意软件，我也对Windows程序有了基本的认识。加上本学期学习汇编语言时编写的游戏，我对Win32编程有了更为深入的理解。本次实验也让我更加熟悉IDA Pro的操作。